

AMENDMENTS TO THE CLAIMS

Claims 1, 5, 13, 16, 24, 27, 30, 35, and 36 have been amended, and claims 37-39 have been added. Thus, claims 1, 2, 4-13, 15-24, and 26-39 are pending in the present application. The following Listing of Claims replaces all prior versions and listings of claims in the present application.

Listing of Claims

1. (Currently amended) A method for generating real-time embedded software code for a wagering game of chance having a randomly-selected outcome, comprising:

preparing an analysis model for the wagering game of chance, the analysis model describing functionality to be included in the software code;

preparing a design model for the wagering game of chance, the design model including a plurality of objects for realizing the functionality in the analysis model, wherein the design model defines static relationships between the objects and dynamic behavior of the objects, wherein the functionality realized is determined by the design model based on the analysis model; and

generating software code for the wagering game of chance from the design model, the software code including at least a portion that is automatically generated in real-time using a software development tool, wherein the automatically generated portion of the software code includes the static relationships between the objects and the dynamic behavior of the objects.

2. (Original) The method of claim 1, wherein the analysis model, the design model, and the software code are prepared using the software development tool.

3. (Cancelled)

4. (Previously presented) The method of claim 1, wherein the design model includes object model diagrams and state charts, the object model diagrams defining the static relationships between the objects, the state charts defining the dynamic behavior of the objects.

5. (Currently amended) The method of claim 1, wherein the functionality described by the analysis model is organized into use cases, the use cases being selected from the group consisting of: handling money, playing the wagering game, handling critical events, and servicing the machine.

6. (Original) The method of claim 5, wherein the analysis model includes use case diagrams and sequence diagrams, the use case diagrams defining relationships between the use cases and external actors outside the game of chance, the sequence diagrams defining a sequence of interactions between the use cases and the external actors.

7. (Original) The method of claim 1, wherein the analysis model and the design model conform to the Unified Modeling Language (UML) standard.

8. (Original) The method of claim 1, wherein the game of chance is a slot reel game including a plurality of symbol-bearing reels that are rotated and stopped to place symbols on the reels in visual association with a display area.

9. (Original) The method of claim 1, wherein the software code includes another portion that is manually prepared.

10. (Original) The method of claim 9, wherein the objects are associated with operations, the manually prepared portion of the software code defining the operations.

11. (Original) The method of claim 1, further including the steps of modifying the design model and automatically modifying the software code in response to modifying the design model.

12. (Original) The method of claim 1, further including the steps of modifying the software code and automatically modifying the design model in response to modifying the software code.

13. (Currently amended) An apparatus for generating real-time embedded software code for a wagering game of chance having a randomly-selected outcome, comprising:

means for preparing an analysis model for the wagering game of chance, the analysis model describing functionality to be included in the software code;

means for preparing a design model for the wagering game of chance, the design model including a plurality of objects for realizing the functionality in the analysis model, the design model utilizing the analysis model to determine the functionality to realize, wherein the design model defines static relationships between the objects and dynamic behavior of the objects; and

means for generating software code for the wagering game of chance in real-time from the design model, the software code including at least a portion that is automatically generated by the generating means, wherein the automatically generated portion of the software code includes the static relationships between the objects and the dynamic behavior of the objects.

14. (Cancelled)

15. (Previously presented) The apparatus of claim 13, wherein the design model includes object model diagrams and state charts, the object model diagrams defining the static relationships between the objects, the state charts defining the dynamic behavior of the objects.

16. (Currently amended) The apparatus of claim 13, wherein the functionality described by the analysis model is organized into use cases, the use cases being selected from the group consisting of: handling money, playing the wagering game, handling critical events, and servicing the machine.

17. (Original) The apparatus of claim 16, wherein the analysis model includes use case diagrams and sequence diagrams, the use case diagrams defining relationships between the use cases and external actors outside the game of chance, the sequence diagrams defining a sequence of interactions between the use cases and the external actors.

18. (Original) The apparatus of claim 13, wherein the analysis model and the design model conform to the Unified Modeling Language (UML) standard.

19. (Original) The apparatus of claim 13, wherein the game of chance is a slot reel game including a plurality of symbol-bearing reels that are rotated and stopped to place symbols on the reels in visual association with a display area.

20. (Original) The apparatus of claim 13, wherein the software code includes another portion that is manually prepared using the generating means.

21. (Original) The apparatus of claim 20, wherein the objects are associated with operations, the manually prepared portion of the software code defining the operations.

22. (Original) The apparatus of claim 13, further including means for automatically modifying the software code in response to modifying the design model.

23. (Original) The apparatus of claim 13, further including means for automatically modifying the design model in response to modifying the software code.

24. (Currently amended) A collection of products for developing a wagering game of chance having a randomly-selected outcome in a common visual programming environment, comprising:

an analysis model for the wagering game of chance, the analysis model describing functionality to be included in the software code;

a design model for the wagering game of chance, the design model including a plurality of objects for realizing the functionality in the analysis model, wherein the design model defines static relationships between the objects and dynamic behavior of the objects, and the functionality realized is determined by the design model based on the analysis model; and

software code for the wagering game of chance, the software code including at least a portion that is automatically generated in real-time from the design model, wherein the automatically generated portion of the software code includes the static relationships between the objects and the dynamic behavior of the objects.

25. (Cancelled)

26. (Previously presented) The collection of claim 24, wherein the design model includes object model diagrams and state charts, the object model diagrams defining the static relationships between the objects, the state charts defining the dynamic behavior of the objects.

27. (Currently amended) The collection of claim 24, wherein the functionality described by the analysis model is organized into use cases, the use cases being selected from the group consisting of: handling money, playing the wagering game, handling critical events, and servicing the machine.

28. (Original) The collection of claim 27, wherein the analysis model includes use case diagrams and sequence diagrams, the use case diagrams defining relationships between the use cases and external actors outside the game of chance, the sequence diagrams defining a sequence of interactions between the use cases and the external actors.

29. (Original) The collection of claim 24, wherein the analysis model and the design model conform to the Unified Modeling Language (UML) standard.

30. (Currently amended) The collection of claim 24, wherein the wagering game of chance is a slot reel game including a plurality of symbol-bearing reels that are rotated and stopped to place symbols on the reels in visual association with a display area.

31. (Original) The collection of claim 24, wherein the software code includes another portion that is manually prepared.

32. (Original) The collection of claim 31, wherein the objects are associated with operations, the manually prepared portion of the software code defining the operations.

33. (Original) The collection of claim 24, wherein the software code is automatically modified in response to modifying the design model.

34. (Original) The collection of claim 24, wherein the design model is automatically modified in response to modifying the software code.

35. (Currently amended) A method for generating real-time embedded software code for a wagering game of chance having a randomly-selected outcome, comprising:

preparing an analysis model for the wagering game of chance, the analysis model describing functionality to be included in the software code, wherein the functionality described by the analysis model is organized into use cases, the use cases being selected from the group consisting of: handling money, playing the wagering game, handling critical events, and servicing the machine;

preparing a design model for the wagering game of chance, the design model including a plurality of objects for realizing the functionality in the analysis model;

generating software code for the wagering game of chance from the design model, the software code including at least a portion that is automatically generated in real-time using a software development tool; and

modifying the design model and automatically modifying the software code in response to modifying the design model.

36. (Currently amended) A method for generating real-time embedded software code for a wagering game of chance having a randomly-selected outcome, comprising:

preparing an analysis model for the wagering game of chance, the analysis model describing functionality to be included in the software code, wherein the analysis model includes use case diagrams, the use case diagrams defining relationships between the use cases and external actors outside the wagering game, and the external actors are selected from the group consisting of: a player, a money handling function, a host, and a random number generator;

preparing a design model for the wagering game of chance, the design model including a plurality of objects for realizing the functionality in the analysis model;

generating software code for the wagering game of chance from the design model, the software code including at least a portion that is automatically generated in real-time using a software development tool; and

modifying the software code and automatically modifying the design model in response to modifying the software code.

37. (New) The method of claim 6, wherein the external actors are selected from the group consisting of: a player, a money handling function, a host, and a random number generator.

38. (New) The apparatus of claim 17, wherein the external actors are selected from the group consisting of: a player, a money handling function, a host, and a random number generator.

39. (New) The collection of claim 28, wherein the external actors are selected from the group consisting of: a player, a money handling function, a host, and a random number generator.